ascom

QVoice
Application note

RTD, FTP and HTTP result interpretation

Document: 60 BS 263562 APFT 1.0/a en

Created by: B. Bögli, T. Scholz, R. Wu

Status: released

Version Overview

Vers.	Date	Author	Modification
1.0	29 Nov 01	R. Wu	First edition
1.0a	8 Mar 02	R. Wu	browser cache

Document Management

Word Text File:	APFT RTD FTP app note 1a
Word Template:	Dok A4_e.dot
Number of Pages:	22
Release:	This document was released following formal review
on:	08 Mar 02
by:	Johan DeBruyne

Contents

1 Introduction

This is an Ascom Qvoice application note, on the interpretation of measurement results for Round Trip Delay (RTD, also called PING) and FTP tests.

The main focus is the impact of Temporary Block Flow (TBF) on RTD results, and the influence of TBF, RTD and file size on FTP transfer tests.

It also explains the differences between a commercial browser, and the QVoice HTTP task and why it is preferable to use the latter in network tests.

2 RTD results

Round Trip Delay measurements form a very important part of the IP data test package. Long RTD implies a degradation of network performance, in terms of throughputs (for TCP based protocol like FTP or HTTP).

Long RTDs can be caused by congestion at some point in the network e.g. at the routers SGSN / GGSN in GPRS networks, or long processing times at these places.

Hence, RTD measurements can reveal network problems, and can also to a great extent predict user data throughput when using acknowledged protocols like FTP or HTTP based on TCP.

However the RTD measurement between, say the air interface and Gi interface of a GPRS network can be influenced by the operations of TBF, which is controlled by the network settings.

2.1 Console PING

The console ping.exe application is based on a Microsoft provided DLL called icmp.dll. This DLL is capable of performing an ICMP echo/reply and gives its users a convenient programming interface to perform their own ping measurements.

This type of Microsoft DLL pings have the following characteristics :

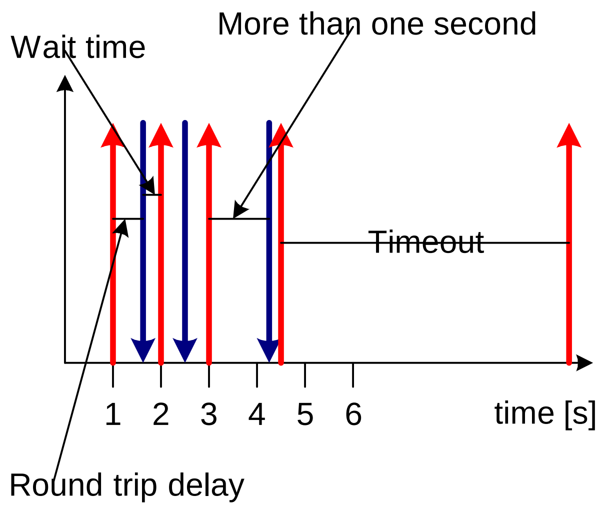
a. not capable of multiple channel from the same computer, i.e. only one channel is usable

b. the time delay between PINGs is fixed to 1 second.

When console PING is used in a GPRS network, the second point will create some subtle effects which impact the measurement result.

2.1.1 Console PING timing

As long as the ping echo returns to the sender in less than 1 second, then the time between PINGs is always 1 second i.e. the wait time between the ping echo reply and the next ping is “one second minus the RTD” :



The first ping (represented by the first red upwards arrow on the left) generated an echo (the blue downwards arrow) within 1 sec, so the second ping is sent one second after the first, and the third ping one second after that (because the second ping's echo reply also comes back within one second).

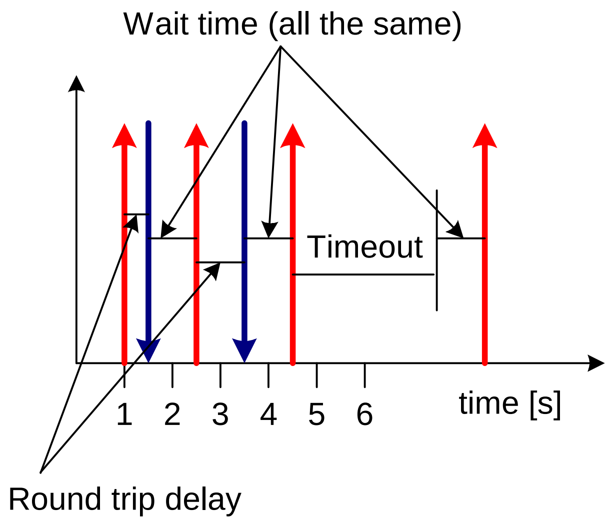
Assume the third ping now has a longer RTD, so the echo reply did not come back within one second (but within the pre-set timeout period). Then console ping will immediately sent the next ping after this echo reply, so there is no wait time.

Another case would be that the RTD is so long that no echo reply comes back and the pre-set timeout is reached. Console ping will now send immediately another ping.

2.2 PING task in QVoice

In order to be able to test more than one channel in one equipment, console ping is not suitable for QVoice, so a new ping feature has been developed by Ascom. Winsockets were used to generate ICMP echo to give the same output like icmp.dll.

Apart from being multi-channel capable, the QVoice ping has another very useful feature : the wait time can be set by the user. (This has the advantage that certain investigations can be done taking the operation of TBF into account. See later)



In the QVoice ping, the user definable wait time is exactly what it says : QVoice waits for that period of time (after the receipt of the echo) before the next ping is sent.

In the diagram, the first ping gets an echo reply within one second, and QVoice starts the wait time counter from the receipt of the echo. On completion of the wait time, the next ping is sent.

QVoice will note the receipt of the echo, and then send the next ping after wait time irrespective of whether the RTD is long or short.

The same principle applies when the RTD is so long that the time out is reached. On reaching time out, QVoice starts the wait time counter and then send the next ping when the wait time is completed

2.3 TBF operation

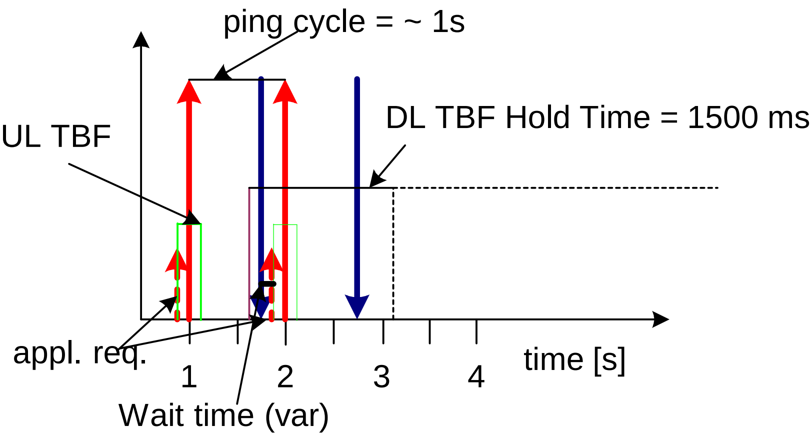
TBF is the operation of allocating radio resources to a subscriber. If a TBF is open for him, it means that he has resources allocated and he can send / receive data. Conversely, a closed TBF means he cannot carry out data transfer.

If the TBF is closed, then he must make a request for radio resources and it will take a little time until the resources is allocated to him and the data transfer can be started.

On the other hand, if he wants to transfer data and there is already an open TBF, then it will be much quicker because he does not need to wait for resource allocation.

2.3.1 Impact of TBF on console ping

Knowing that console ping has a fixed cycle of 1 second, one can see the following situations.



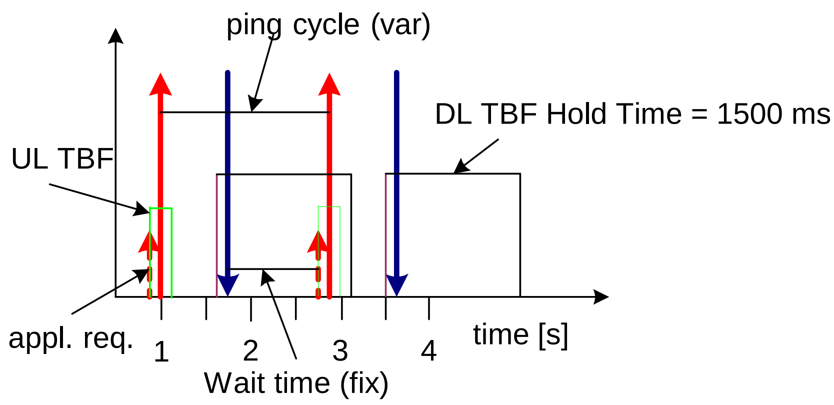
Assuming the pings were sent uplink, and the echo came back downlink. The diagram shows a request for resources (appl. Req. in the diagram) just before the first ping was sent. If now the echo reply came back after one second, then console ping would immediately send another ping uplink.

The fact that there is an echo coming back downlink meant that a downlink TBF was opened. Now if the network setting was such that a downlink TBF, once opened, will stay open for 1500 ms (i.e. a downlink TBF hold time of 1500ms), then the next echo, if it returned within 1500 ms of the first echo, will find that there was already a downlink TBF open and ready. So the second echo did not have to wait for a downlink TBF to open, and hence the second RTD would be shorter (compared to the case when no TBF was open at the time).

So the RTD result can be influenced by the availability of an already opened TBF. This, in turn, is affect by the TBF hold time, and the time between pings.

2.3.2 TBF and QVoice ping

Since the QVoice ping is sent (RTD + wait time) after the receipt of an echo reply, whether or not TBFs are open has an impact on the result.



Assume the QVoice user has set the wait time to 1 second, and the downlink TBF hold time is, again, 1500 ms.

The first ping (at second 1) got an RTD of, say 0.8 second i.e. the echo arrived at second 1.8 on the time axis. QVoice waited for the user-defined wait time of 1 second, and so at second 2.8, the second ping was sent.

The first echo reply caused a downlink TBF to be opened, and it was closed at 1.8 sec + 1500ms = 3.3 second.

The echo reply to the second ping, when it came back, found that the TBF was already closed and had to request for a new one. This caused the second RTD to be longer.

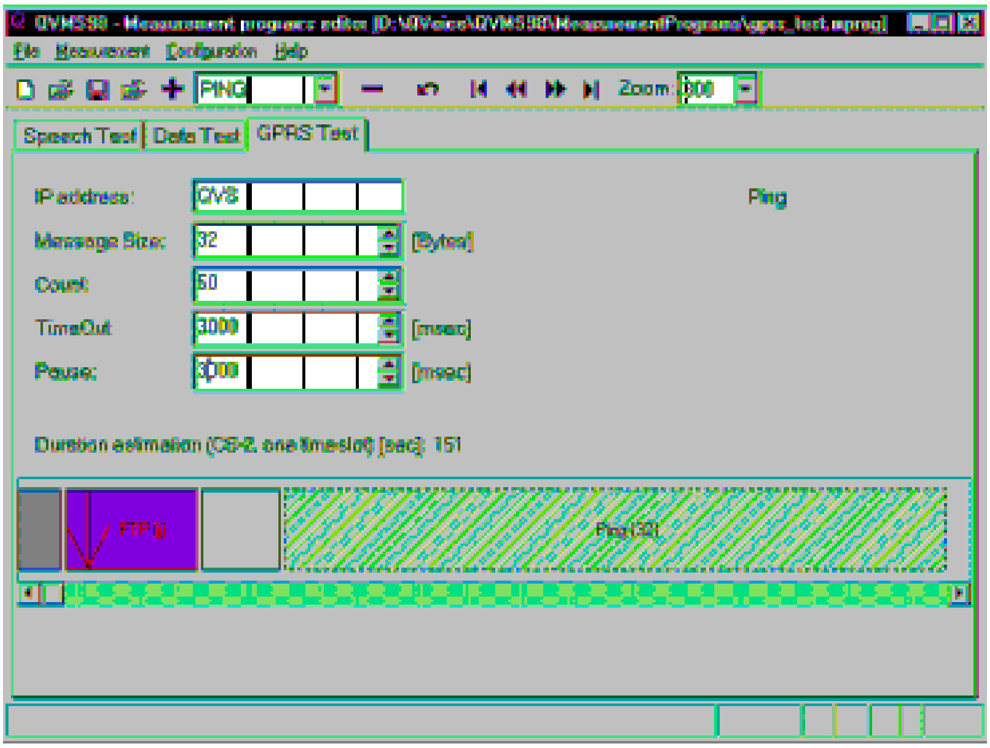
Hence, it can be seen that the console and QVoice pings can give different results, because of the time between pings and the availability of an open TBF.

2.4 Using QVoice ping

By using the user-definable wait time, the QVoice user can investigate different issues in a network.

2.4.1 TBF hold time

By varying the QVoice ping wait time (relative to the TBF hold time), one can verify whether the TBF is operating as one expect.



For example, if the wait time (pause) is very short (say 100 ms), and the TBF hold time is comparatively long, then one would expect the RTD measurements to give rather short RTD values.

2.4.2 Latency

The time it takes for the network to deliver a reaction can be tested. One must be careful that no open TBF is available to generate a false measurement.

Hence, one can set the QVoice ping wait time to be very long (e.g. 10 seconds). This would ensure that after the first ping and its echo, the TBFs will be closed (during this 10 second wait).

The next ping will then give the RTD / latency measurement showing the reaction time without the help of any already opened TBF.

3 FTP results

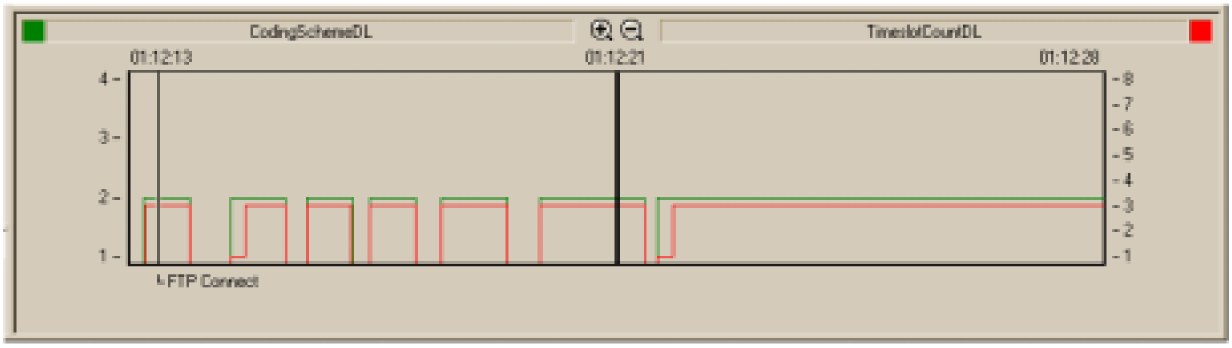
FTP measurements may seem straightforward at first glance, but on closer inspection, there are issues to bear in mind when interpreting the results.

3.1 Impact of TBF lengths

There is a compromise to be reached by the network TBF settings. Giving long TBF (i.e. keeping TBFs open for a long time) to a subscriber will ensure higher data throughput. However, doing so will mean that radio resources could be wasted if he is not using the open TBF all the time.

3.1.1 Short TBF allocation

This real life measurement shows the network assigning short TBFs to the subscriber at the start of the FTP transfer (FTP connect marker) :

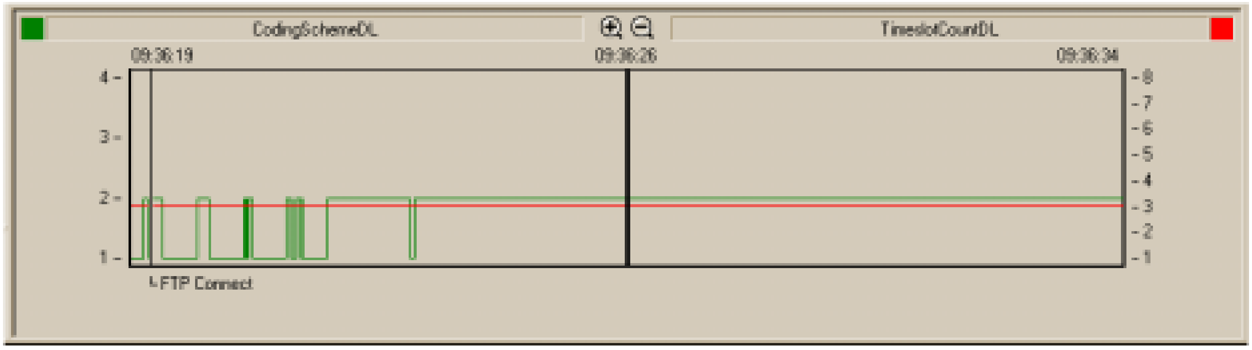


A short TBF (marked by red lines between 1 and 2 on the left hand scale) was opened and then quickly closed. When it was closed, the user has still a large amount of data to be transferred so another TBF has to be opened. This was also a short one, and another one was needed and so on. Eventually, a long TBF was allocated and the user can transfer data at a higher speed (without requesting and waiting for TBFs to open).

This type of TBF allocation of course will lower the FTP user data throughput rate. In fact, the throughput rate would be slow at the beginning, and rising towards a higher value later on.

3.1.2 Long TBF

In contrast, another network measurement showed that it gave the user a long TBF right from the start of the TBF transfer :



The long TBF was shown by the horizontal red line extending to the right. All other things being equal, this TBF allocation would give a higher FTP throughput than the short TBF case above.

3.2 The impact of TCP

Since FTP uses, in turn, the TCP protocol, the operation of TCP also has an impact on the FTP throughput rate.

When TCP starts to transfer user data, the first buffer of user data sent would be very short. If the receiver acknowledges the correct receipt of this short data, TCP would now send a longer buffer. If that is again correctly received, it will send a longer one still. (If there is no positive acknowledgement, TCP would re-transmit to correct the error).

This process is described in RFC 793, but the exact details can be implementation dependent (i.e. Windows implementation may be different from stacks from other manufacturers).

Hence, the operation of TCP will ensure that the user data throughput rate would be slow to start with, and rises to a maximum later on.

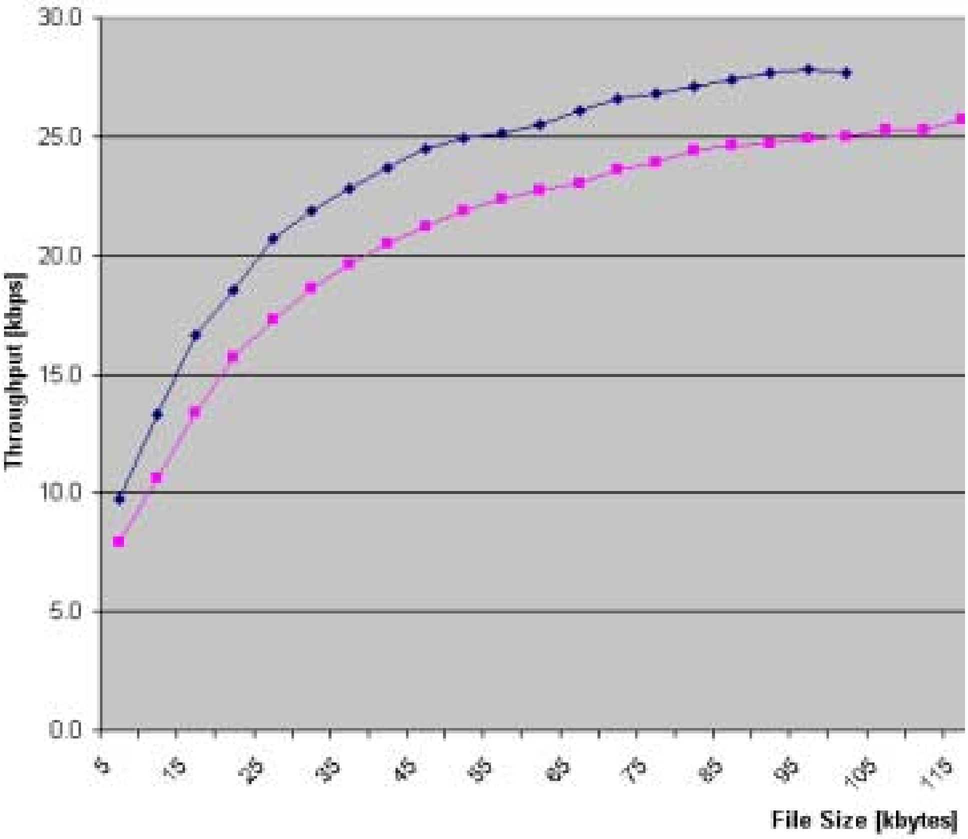
3.3 The effect of file size on throughput

In the previous discussion, there were at least two reasons why the FTP throughput could be low to start off with, and rises to a higher value later :

- the effect of short TBF at the beginning
- the effect of TCP building up longer buffers of data

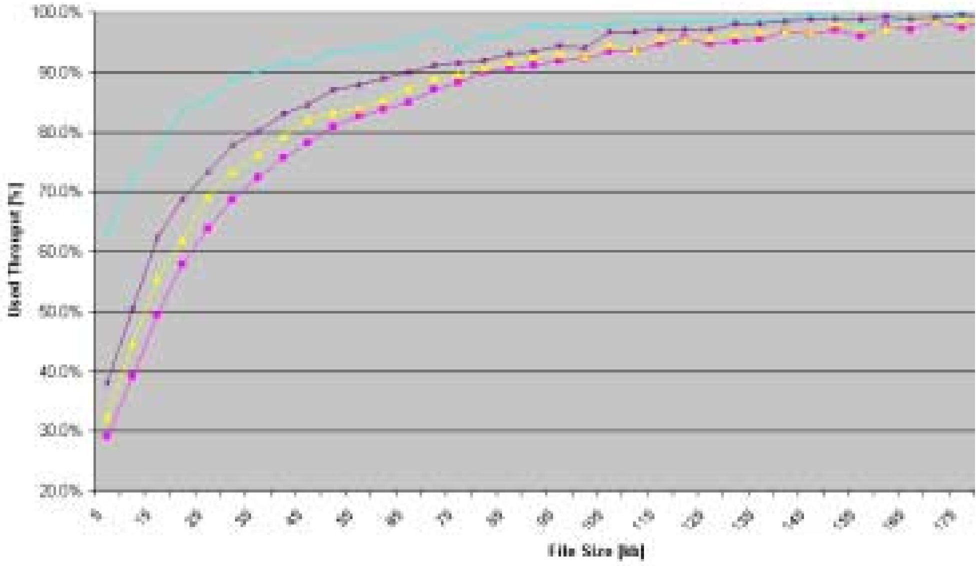
This implies that if the file to be transferred is short, then this build up of throughput will cause the overall user data throughput rate to be lower, when compared to a longer file.

One can see this in the measurements (with Windows NT operating system) below :



FTP transfers of files of different sizes were made via two GPRS networks. One network actually reaches a maximum FTP throughput of around 28 Kbps for longer files, while the other gives a maximum of 26 Kbps. While the absolute value of throughput differs slightly, the shape of the two curves are the same ----- the throughput reached rises with the file size.

If one now plots the values normalised to the maximum throughput (i.e. the maximum throughput of each network is shown as 100%), then we have the following :



The top curve is FTP transfer via ISDN i.e. the effect of TBFs was not present. It reaches 90% of its maximum throughput value for a file about 38 Kbytes, which means that for files smaller than this size, the FTP transfer was completed before the maximum throughput can be reached.

The second curve from the top is FTP via HSCSD. Again there are no effects of TBF. So the throughput is dictated by the TCP buffer build up process, and the RTD in HSCSD. The TCP process would be similar to the ISDN case, but the RTD is longer. So it took a file of about 70 Kbytes to reach the 90% maximum throughput.

The two lower curves are for two different GPRS networks. Here, the effects of TBF, TCP and RTD combines to make the curve flatter than the previous two. It took files of 85Kbytes to reach 90% maximum throughput.

3.4 Interpretation of FTP measurements

One can now see that "FTP throughput" measurements can give different results depending on :

- the RTD of the network (which can vary according to traffic load at the time)
- the TBF allocation algorithm used
- the file size used for the measurement
- the coding scheme assigned by the network
- the timeslots available during the transfer (can vary according to traffic load)

- the implementation of the TCP / RFC 793 process in that particular stack
- etc.

Care should be taken, therefore, to compare FTP throughputs measured even in the same network, and of course in different networks. The conditions in which the measurements were carried out should be carefully examined before comparing the results.

4 HTTP results

HTTP is the protocol for accessing information on a Web page. It commonly uses TCP and therefore the discussion on TCP in the previous chapter also applies.

In addition, there is the issue of using commercially available browsers (e.g. Netscape, Internet Explorer etc.) and how it might affect the measurement result.

4.1 Commercially available browsers

These are relatively complex software carrying out two main tasks : access and download the desired information, and interpret /display the information on screen.

They are somewhat complex because they attempt to carry out several tasks in parallel. A Web page typically consists of a number of small files, e.g. some text and some pictures. These commercial browsers will attempt to do several things simultaneously :

- download several files in parallel, and
- interpret and display them in parallel, sometimes simultaneously with the download process
- store previously accessed Web page in its local cache memory, in an attempt to reduce access time if the same page is accessed again

This method of working has several consequences :

a. since each commercially available browser would carry out these simultaneous tasks in a different way, the browser performance becomes a factor in, say, throughput measurements. Different browsers will give different measurement results.

b. Even with the same browser, the way it works can be different from one software release to another. Since QVoice's objective is to measure the cellular network (and not the browser performance) if some measurements were done with release x of a specific browser, this result may or may not be compared to measurements done with release x + 1 ---- i.e. consistency with historic data cannot be guaranteed.

c. Most of these commercially available browsers can only work with one communication path i.e. benchmarking several networks from one system is not possible.

d. The exact way how it works is known only to the manufacturer of the browser, and is not publicly available. So browser users are not aware of how it carries out its functions, and what impact it may or may not have on measurement results.

e. Repeatedly accessing the same Web page will trigger the commercial browsers to load the page from its local cache memory. This means that the load time measurement is not related to the network performance, but to the cache memory size/speed etc. (see later)

4.2 QVoice HTTP measurements

The objectives of QVoice HTTP measurements are :

a. to measure the performance of the cellular network (not the interpret / display function of a browser nor the performance of its cache memory)

b. to keep the method of HTTP measurements the same over different software releases, so historic data comparisons can be made. Differences between current and historic data will be due to the performance of the cellular network, and not factors like different browser software performance

c. simultaneous testing of several cellular networks with HTTP protocol is possible

d. a clear / well defined way of working

To achieve these goals, Ascom has specially developed a HTTP measurement which fulfils these objectives.

4.3 Comparing commercial browser and QVoice HTTP

It is clear from the above that the QVoice HTTP measurements should not be compared to those of the commercial browser. This is because :

a. the method of working is different. QVoice HTTP task does not attempt to do several things in parallel as in a browser.
b. A browser will attempt to interpret / display the information from the Web page download, whereas QVoice concentrates in the download part, and not the interpret / display functions.
c. QVoice HTTP ensures that the network performance is measured, and not the browser/cache performance.

4.3.1 Impact of cache memory in commercial browsers

If one repeatedly access the same Web page, and compares the download times achieved by a commercially available browser (e.g. Internet Explorer or Netscape) to that of QVoice HTTP, one could find that the commercial browsers would show shorter download times.

Assuming the purpose of the measurement is to test the performance of the tranmission network, and not the browser performance, then it is not advisable to use the commercial browsers.

The reason is that commercial browsers would store previously accessed page(s) in its local cache memory. If the same page is requested again and the browser finds that it is available in its cache, it will access the Web page (via the tranmission network) to check that the page size etc. has not changed. If this is the case, then it will download the contents of the page, not from the distant server via the transmission network, but from its local cache.

This leads to two issues :

- the download time will be much shorter (than loading the entire content from the distant server via a transmission network e.g. GPRS)
- the performance of the transmission network is not measured at all.

Therefore, QVoice uses its own HTTP tests, in order to ensure that it is really measuring the transmission network performance (and not a local cache).

An interesting measurement would be to compare the download time of commercial browsers WITHOUT its cache memory, to that of QVoice HTTP.

4.3.2 Commercial browser without cache / QVoice HTTP

The following measurements were carried out with different browsers installed on a PC with the following main characteristics :

- Pentium III 500 MHz,
- 250 MB RAM,
- Operation System : Windows NT

Access were made to the internet over a GPRS network with :

- Good RF coverage
- User was stationary (not moving in a car or walking)
- Mobile phone : Sagem OT96

The target Web page was in a webserver installed in public internet.

The same webpage has been downloaded many times with different browsers in this same environment.

It is very important to ensure that the browser doesn't use any cache memory. For this reason we took care to switch off all cache memories, as follows.

For the Internet Explorer (shown as IE in result table below) :

Version : MS Internet Explorer 5.5

Measurement procedure :

1. Start of IE

2. Select : Tools -→ Internet Options -→ Temporary internet files -→ Delete Files (select OK)
3. Select URL
4. Downloadtime was measured
5. Stop IE

go to 1 and repeat the same procedure for every measurement i.e. the Internet Explorer was started again for each measurement.

For Netscape (shown as NC in result table below) :

Version : Netscape Communicator 4.7.9

Measurement procedure :

1. Start NC
2. Select : Edit -→ Preferences -→ Advanced --→ Cache :
3. set memory cache = 0 and select : clear
4. set disk cache = 0 and select : clear
5. Select URL
6. Downloadtime was measured
7. Go to 5 Select URL …..

For QVoice HTTP tests :

QVoice HTTP application release 3.0, automatic measurement procedure with 20 s pause in between HTTP download

Four different webpages of different sizes have been downloaded from the same server under conditions as described.

Each webpage has been downloaded 30 times. This number is high enough to reach at least a 10% confidence interval with 95% confidence level.

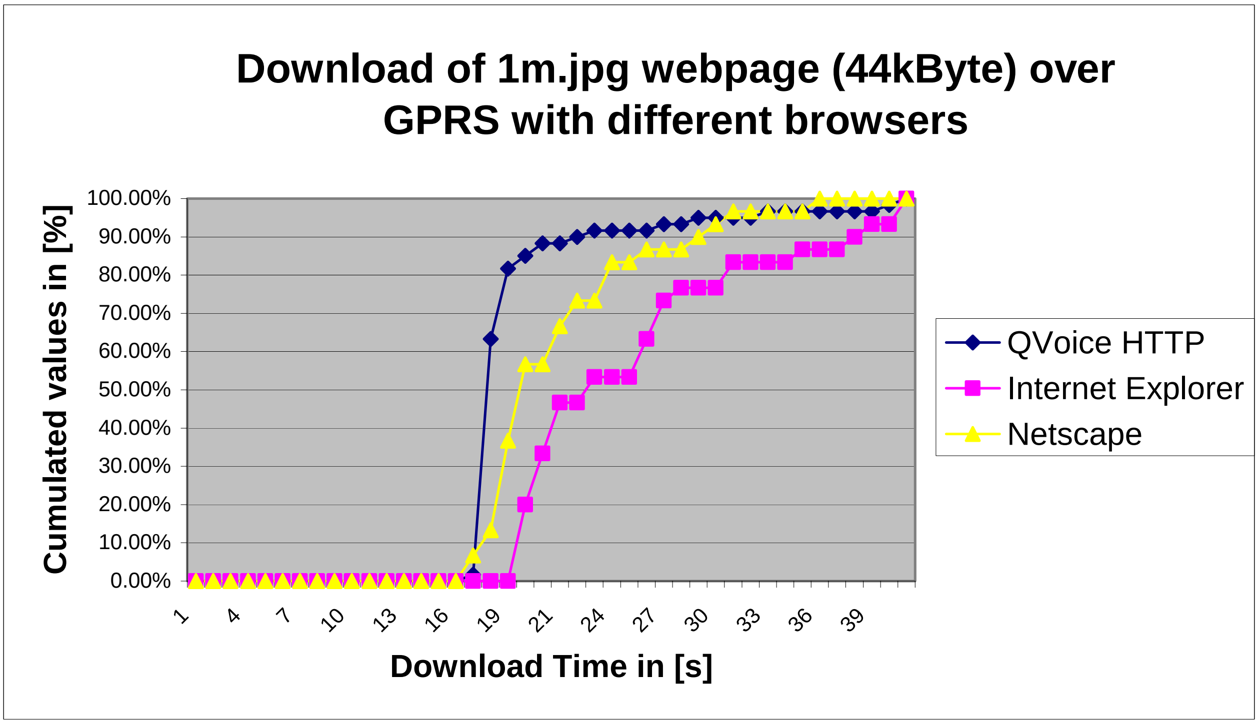
4.3.3 Measurement result overview

		QVoice		**IE**		**NC**	
filename	size [kByte]	**Time [s]**	StDev	**Time [s]**	StDev	**Time [s]**	StDev
1m.jpg	44.7	**18.6**	5.3	**25.2**	7.8	**20.6**	4.8
2m.jpg	38.6	**15.8**	1.9	**21.1**	5.8	**19.2**	6.8
1sm.jpg	10.4	**6.1**	2.4	**10.3**	3.8	**6.4**	3.2
2sm.jpg	10.5	**5.9**	1.9	**13.5**	20.6	**5.7**	2.1
avg		**11.6**		**17.5**		**13.0**	

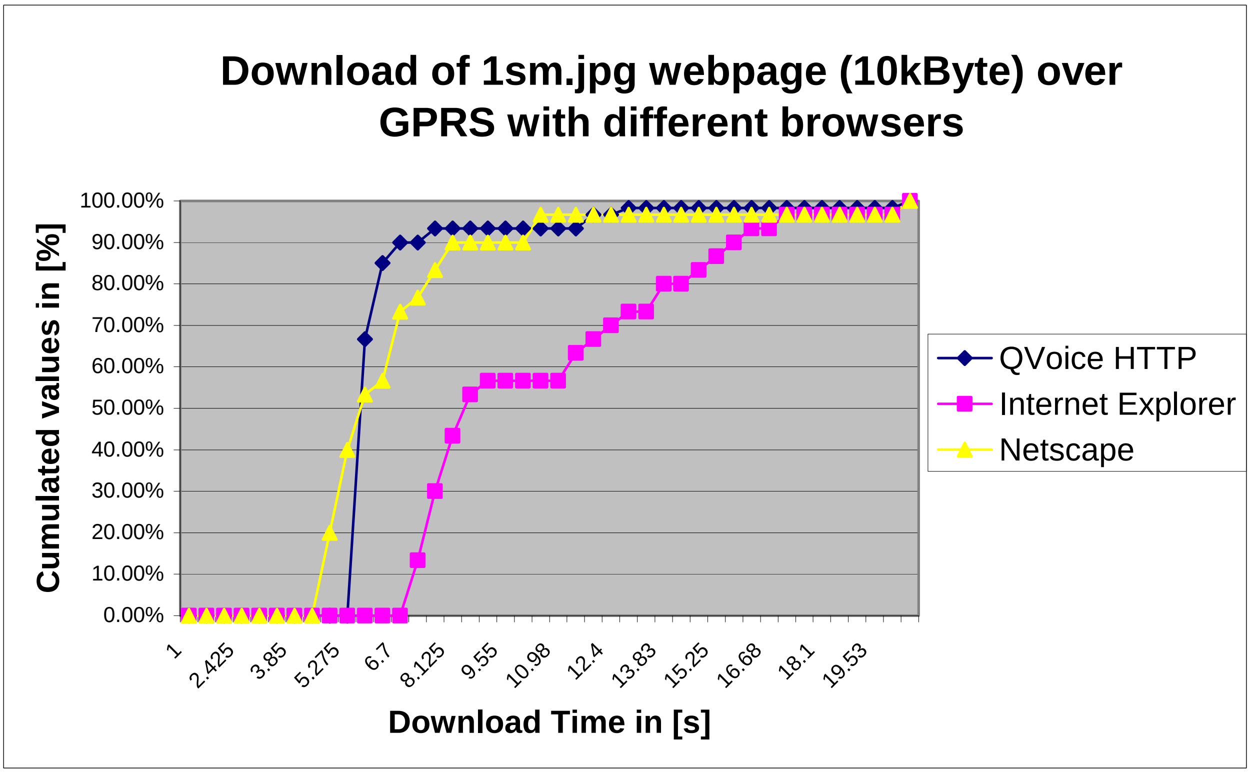
StDev = standard deviation, although the distribution is not exactly Gaussian, the StDev value gives a good idea about the spreading in the results.

The distribution of the results is shown below for 2 different webpages (1m.jpg and 1sm.jpg)

4.3.4 Download of 1m.jpg webpage



4.3.5 Download of 1sm.jpg webpage



It can be seen that there are no significant differences between the QVoice HTTP download times compared to those of the commercial browsers without cache. If anything, the QVoice HTTP is faster.

5 Conclusions

Due to the complex interaction of

TCP process

TBF allocation algorithm

round trip delay

traffic conditions in the network at measurement time

coding scheme allocations

time slot availability

interference / error rate

etc.

the measurements of FTP throughput (and also for HTTP) and RTD need careful planning before the measurements, and the results need some in-depth analysis in order to understand the impact of different network environment.

For some investigation work, the use of PINGs of different sized/ different timing and the use of UDP (instead of TCP based protocols like FTP) can give clearer results.

The QVoice HTTP task has the advantages of consistency over different software releases, mutli-channel capabilities and a method of working which is precisely known. Because it has a different objective and works in a different way to a commercial browser, the QVoice HTTP results should not be compared to those from a commercial browser.

Doing repeated downloads of the same Web page will introduce the issue of local cache memory in the commercial browsers. Since the objective is to measure transmission network performance, one should not use the commercial browsers for this purpose.